Computer Systems II

Gordon College

Operating System Overview

Class Intro

- Operating System Class
- Two Directions:
  - Practical
    - Linux+ Guide to Linux Certification and Lab Manual
    - Lab Experience
  - Theoretical
    - Operating Systems (3rd Edition Gary Nutt)
    - Lecture and Projects

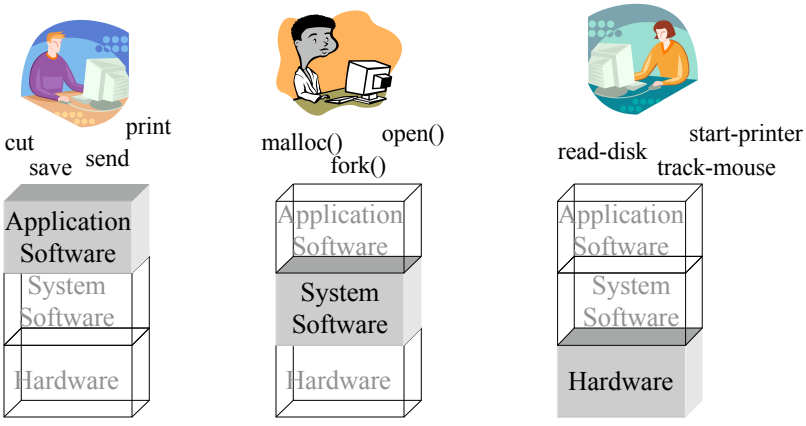
Why Study Operating Systems?

Slide 1-3

- Understand the *model of operation*
  - Easier to see how to use the system
  - Enables you to write efficient code
- Learn to design an OS
- Even so, OS is pure overhead of real work
- Application programs have the real value to person who buys the computer

Perspectives of the Computer

Slide 1-4



(a) End User View

(b) Application Programmer View

(c) OS Programmer View

Slide 1-5

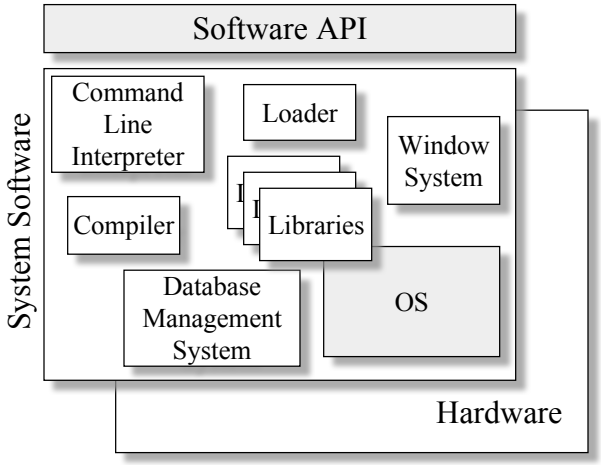
System Software

- Independent of individual applications, but common to all of them
- Examples
  - –C library functions
  - –A window system
  - –A database management system
  - –Resource management functions
  - –The OS

Slide 1-6

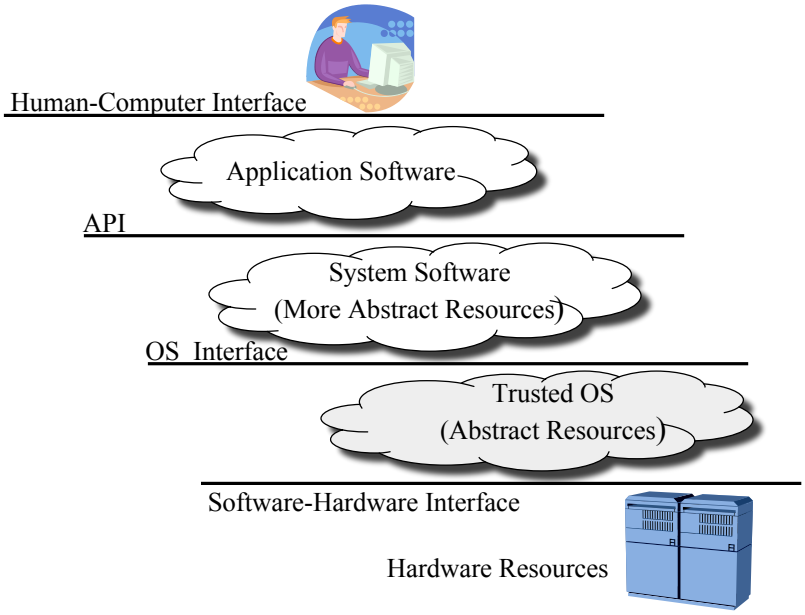
Using the System Software

Application Programmer



Application Software, System Software, and the OS

Slide 1-7



The OS as Resource Manager

Slide 1-8

- Process: An executing program
- Resource: Anything that is needed for a process to run
  - –Memory
  - –Space on a disk
  - –The CPU
- "An OS creates resource abstractions"
- "An OS manages resource sharing"

Resource Abstraction

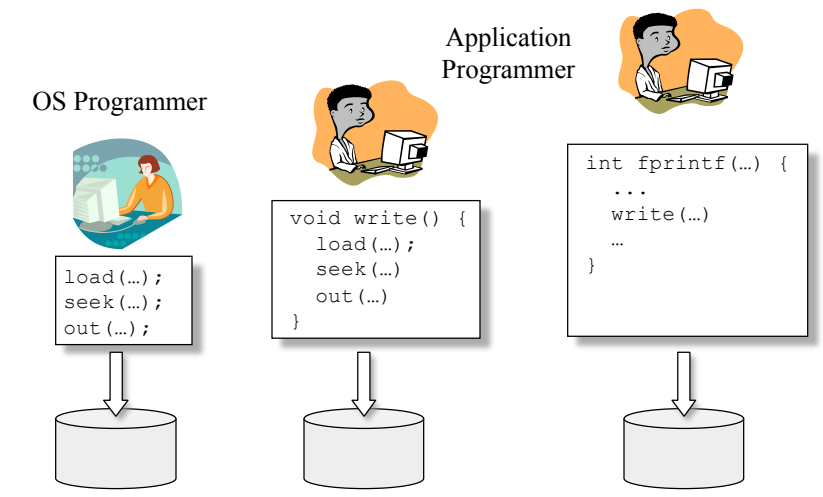

```
load(block, length, device);
seek(device, 236);
out(device, 9)
```

```
write(char *block, int len, int device,
                  int track, int sector) {
   ...
  load(block, length, device);
  seek(device, 236);
  out(device, 9);
   ...
}
```

```
write(char *block, int len, int device,int addr);
```

```
fprintf(fileID, "%d", datum);
```

Disk Abstractions



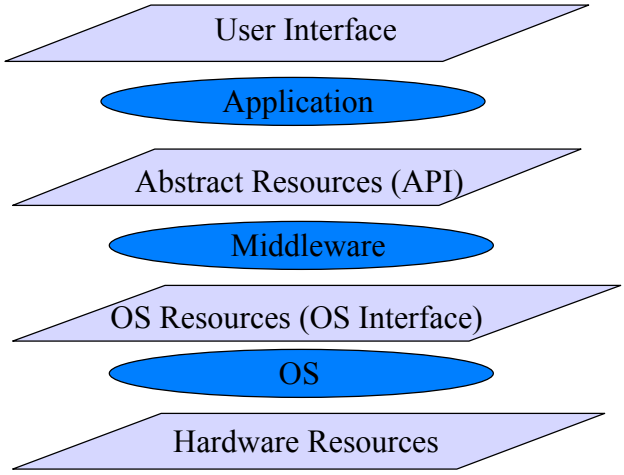
(a) Direct Control

(b) `write()` abstraction

(c) `fprintf()` abstraction

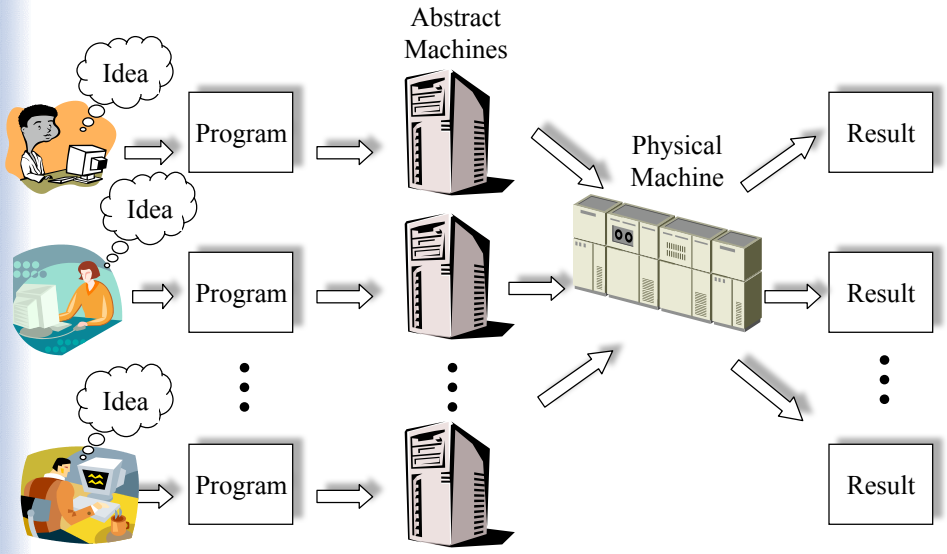
Slide 1-11

Abstract Resources



Slide 1-12

Abstract Machines



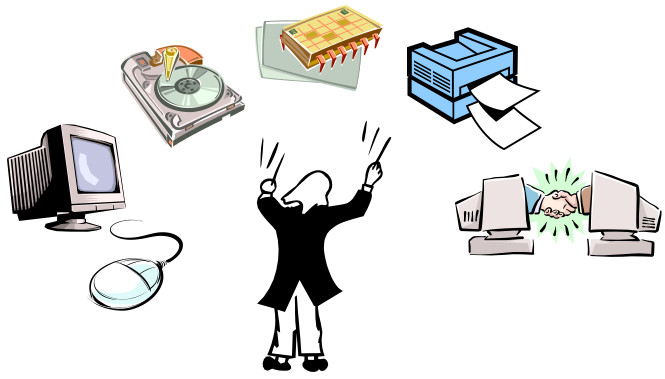
Resource Sharing

Slide 1-13

- Space- vs time-multiplexed sharing
- To control sharing, must be able to *isolate* resources
- OS usually provides mechanism to isolate, then selectively allows sharing
  - How to isolate resources
  - How to be sure that sharing is acceptable
- Concurrency

The OS as a Conductor

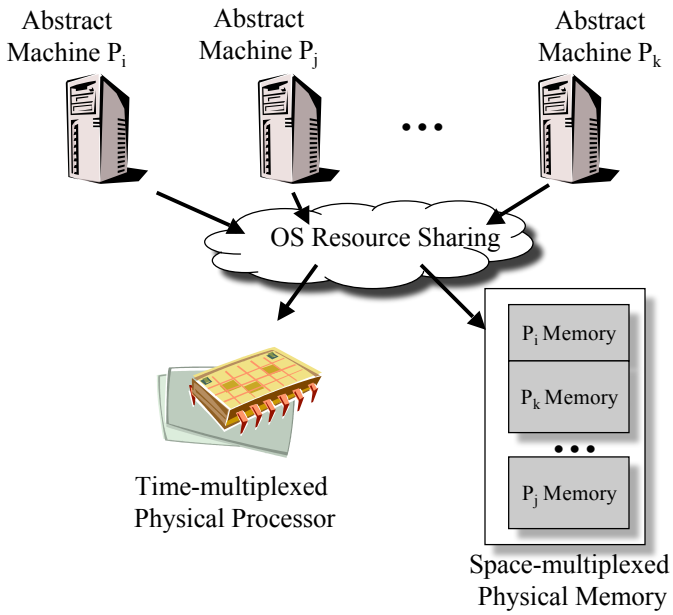
Slide 1-14



The OS *coordinates the sharing and use* of all the components in the computer

Multiprogramming

Slide 1-15



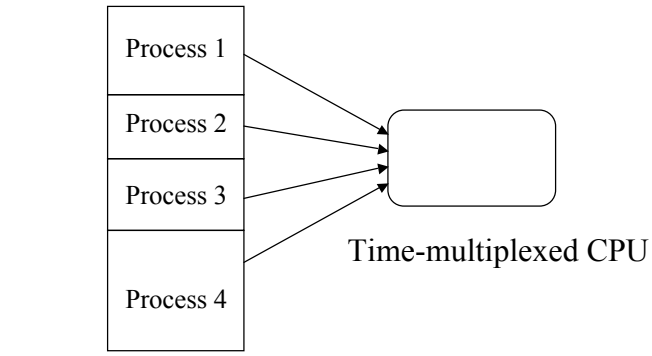
Slide 1-16

Multiprogramming(2)

- Technique for *sharing* the CPU among *runnable* processes
  - Process may be *blocked* on I/O
  - Process may be *blocked* waiting for other resource, including the CPU
- While one process is blocked, another might be able to run
- Multiprogramming OS accomplishes CPU sharing "automatically" – *scheduling*
- Reduces time to run all processes

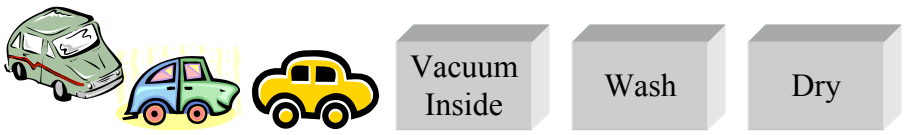
How Multiprogramming Works

Slide 1-17

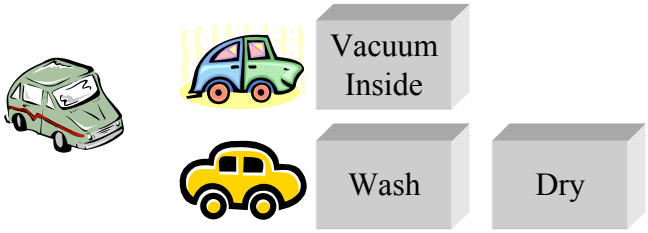


Speeding Up the Car Wash

Slide 1-18



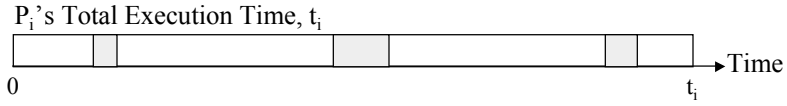
(a) The Sequential Car Wash



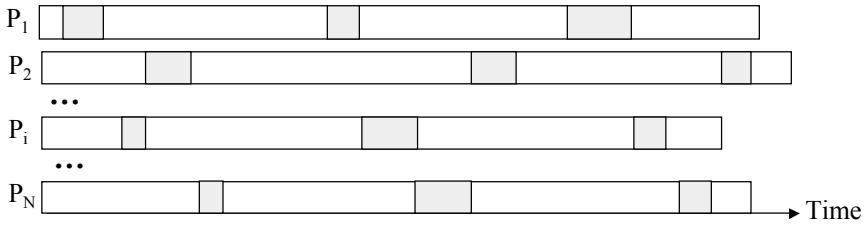
(b) The Parallel Car Wash

Multiprogramming Performance

Slide 1-19



(a) P_i's Use of Machine Resources



(a) All Processes' Use of Machine Resources

Using the processor
I/O operation

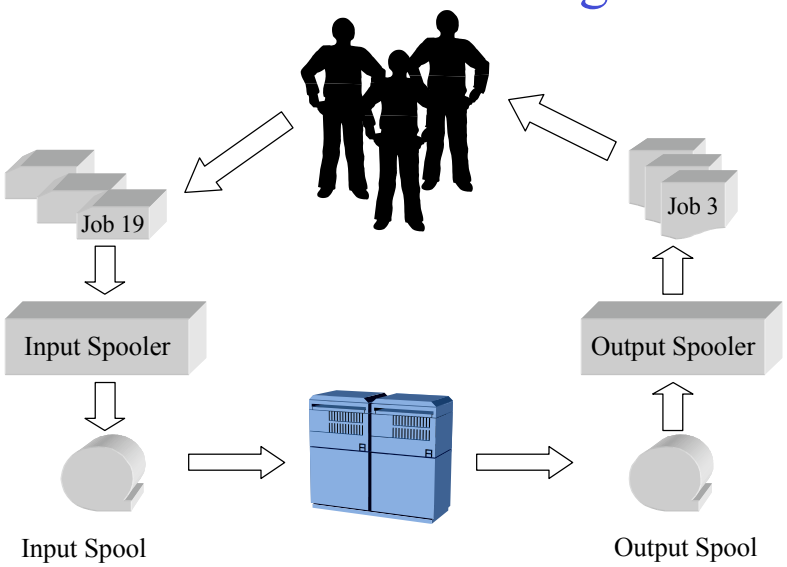
OS Strategies

Slide 1-20

- Batch processing
- Timesharing
- Personal computer & workstations
- Process control & real-time
- Network
- Distributed
- Small computers

Slide 1-21

Batch Processing



Slide 1-22

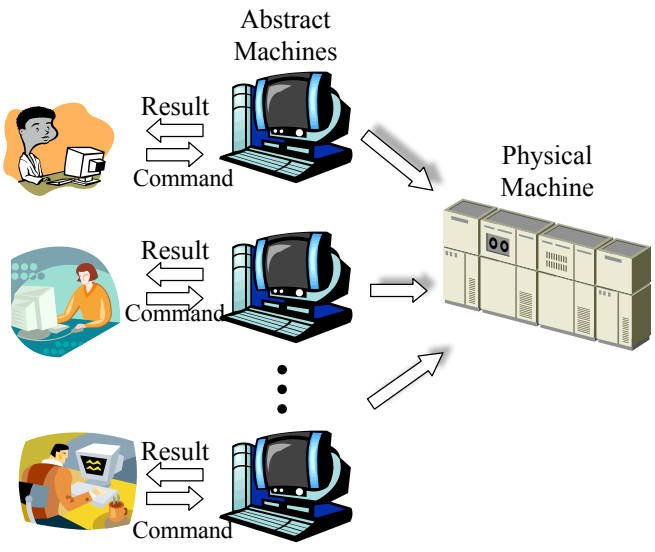
Batch Processing(2)

- Uses multiprogramming
- *Job* (file of OS commands) prepared offline
- Batch of jobs given to OS at one time
- OS processes jobs one-after-the-other
- No human-computer interaction
- OS optimizes resource utilization
- Batch processing (as an option) still used today

A Shell Script Batch File


```
cc -g -c menu.c
cc -g -o driver driver.c menu.o
driver < test_data > test_out
lpr -PthePrinter test_out
tar cvf driver_test.tar menu.c driver.c test_data test_out
uuencode driver_test.tar driver_test.tar >driver_test.encode
```

Timesharing Systems



Timesharing Systems(2)

Computer Research Corp '67

- Uses multiprogramming
- Support interactive computing model (Illusion of multiple consoles)
- Different scheduling & memory allocation strategies than batch
- Tends to propagate processes
- Considerable attention to resource isolation (security & protection)
- Tend to optimize response time

Personal Computers

- CPU sharing among one person's processes
- Power of computing for personal tasks
  - Graphics
  - Multimedia
- Trend toward very small OS
- OS focus on resource abstraction
- Rapidly evolved to "personal multitasking" systems

Process Control & Real-Time

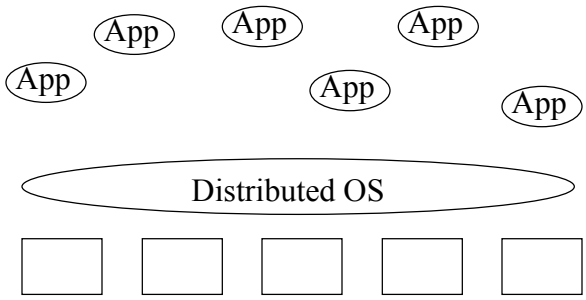
- Computer is dedicated to a single purpose
- Classic embedded system
- Must respond to external stimuli in fixed time
- Continuous media popularizing real-time techniques
- An area of growing interest

Networks

- LAN (Local Area Network) evolution
- 3Mbps (1975) → 10 Mbps (1980) → 100 Mbps (1990) → 1 Gbps (2000)
- High speed communication means new way to do computing
  - Shared files
  - Shared memory
  - Shared procedures/objects
  - ???

Distributed OS

- Wave of the future



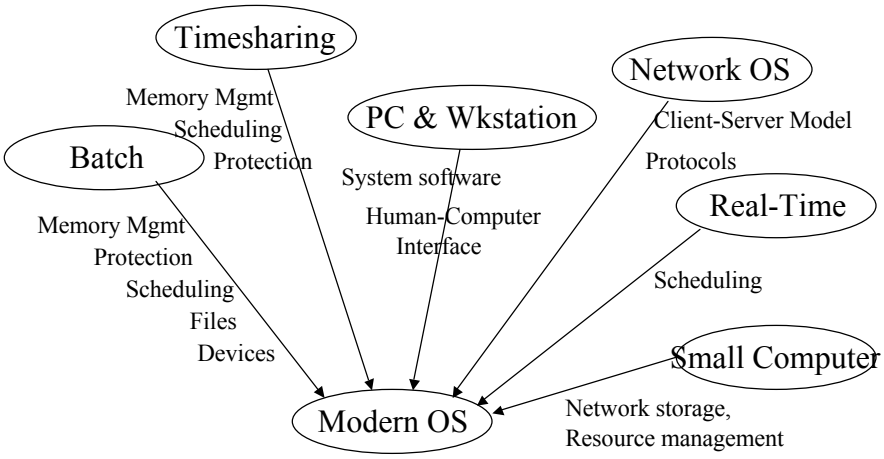
Multiple Computers connected by a Network

Small Computers

- PDAs, STBs, embedded systems became commercially significant
- Have an OS, but
  - Not general purpose
  - Limited hardware resources
  - Different kinds of devices
    - Touch screen, no keyboard
    - Graffiti
  - Evolving & leading to new class of Oses
- PalmOS, Pocket PC (WinCE), VxWorks, …

Evolution of Modern OS

Slide 1-31



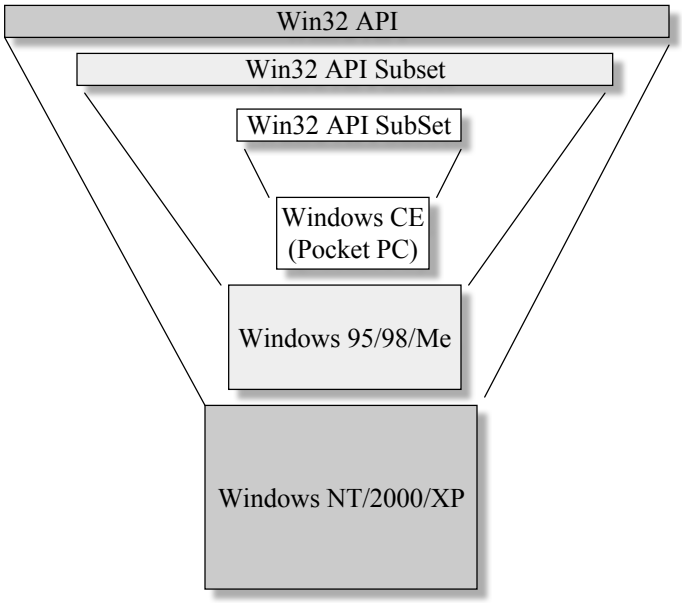
Examples of Modern OS

Slide 1-32

- UNIX variants (e.g. Linux) -- have evolved since 1970
- Windows NT/2K -- has evolved since 1989 (much more modern than UNIX
  - Win2K = WinNT, V5
- Research OSes – still evolving …
- Small computer OSes – still evolving …

The Microsoft OS Family

Slide 1-33



Summary

Slide 1-34

An Operating System must be able to:

- provide functionality to apps
- provide abstraction of hardware to users and apps
- provide the sharing of resources to processes
- provide security and protection
- be as transparent as possible
- be as light as possible